Interfacing training content with simulations/simulators.

Summary
There are many instructional paradigms that involve simulations and simulators to provide higher-order "Learn by Doing." SCORM, for the most part, addresses concerns only for self-paced, web-delivered training. There are several epistemological and technical problems with interfacing this type of training with simulations and simulators. However, one of the largest epistemological concerns with using simulators in self-paced training, is that there is a great chance of "negative learning." That is, without instructional supervision, a learner may learn the wrong thing: From false assumptions and incorrect procedures; to dangerous shortcuts and hazardous habits.

Requirements/Needs
There is a need to be able to incorporate simulations and simulators into SCORM and AICC content. The method of incorporation needs to be done in a way that is familiar and understandable to SCORM and AICC content developers. These developers need to be able to communicate with simulations. Since they must also communicate with learning management systems, it is logical to use similar, if not identical methods. Using the same method will provide new functionality without requiring new skills.

There is also a need to incorporate training content into simulations or have it available to the learner in the simulation if needed. There are several use cases for having content available for the learner in a training device. For example, presenting content before an exercise can provide context and help cognitively immerse the learner into the exercise. Another example is presenting instructional information as required during the exercise to help a learner understand the concepts required. This can take the form of on-demand refresher information, cognitive aids (charts, graphs, tables, and memory aids), or remedial training.

Recommendations
The recommendation is to use the same API developers use to communicate with the LMS to communicate with simulations. That is, find the simulation's API then use "setValue" and "getValue" to initialize, monitor, and control the simulation.

The API will require minor changes. For example, the "initialize" command should take an argument specifying the scenario for the particular learning activity. There also needs to be a method of indicating a callback function for the simulation to call when an event occurs. While the training content can continuously poll for values, it is much more efficient for the content to subscribe to events or data. Ideally, both would be supported. For example, *setEvent_Listener("engineState")* to be notified when the learner starts or stops

the engine (changes the engine's state) and *setData_Stream("speed", "10 hz")* to be automatically notified the speed of the vehicle at a rate of ten times a second.

A taxonomy of simulation interfaces needs to be developed similar to the CMI taxonomy. This taxonomy would include data elements to control the simulation, such as simulator modes (e.g. run, stop, pause), speed (e.g. 1x, 2x, .5x), indicate scenario (initialization) files or modes, and to set restrictions or overrides such as turning off inputs from specific simulator panels.

Efforts to create a specification or standard has been started a couple of times by the Aircraft Industry CBT Committee (AICC), at least once by IMS Global Learning Consortium, twice by the Simulation Interoperability Standards Organization (SISO), by the DoD's DARWARS initiative, and by the ADL Joint Co-Lab. However, each of these efforts took a narrow, limited approach or could not find enough champions for support.

The last effort by the SISO SCORM/Simulation Interoperability Study Group should have been the most successful since most of the burden of implementing the interfaces will be on the simulation developers who support and participate in SISO. This effort was originally supported by many content developers and, more importantly, ADL, AICC, and IEEE's Learning Technology Standards Committee (LTSC). However, it failed to get the support of the simulator development companies and it failed to agree on an approach.

The recommendation to make this standard is to proceed through the SISO with coordinated support from LETSI, AICC, and LTSC. Or through LETSI with coordinated tasks by each of the groups. For example, a SISO group would define the taxonomy and interface with the scenario standards groups, while an AICC group would work on the API, LTSC would develop the metadata and LETSI would develop the use cases and oversee the work.

Brandt W. Dargue, ATF
Boeing Associate Technical Fellow
Training Technologies
Phantom Works Support Technologies - Logistics
mailto:Brandt.W.Dargue@Boeing.com